

### REMARKS/ARGUMENTS

Claims 1-39 are pending in the application. Claims 1, 2, 5, 6, 12, 14, 15, 18, 19, 25, 27, 28, 31, 32, and 38 have been amended. Reconsideration is respectfully requested. Applicants submit that the pending claims 1-39 are patentable over the art of record and allowance is respectfully requested of claims 1-39.

The Examiner has objected to a previously filed Information Disclosure Statement. Applicants are re-submitting an Information Disclosure Statement with this amendment to overcome the objection.

Claims 1-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Melton et al. (U.S. Patent No. 6,289,336). Applicants respectfully traverse.

Anticipation requires that the identical invention must be shown in a single reference in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claims 1, 14, and 27 describe, under control of an iterator function processor that executes as part of a data store engine (e.g., Specification, page 6, paragraph 18, FIG. 1), when an iterator function included in a statement is invoked, obtaining one or more predicates included in the statement, applying the one or more predicates to a row of data, if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match.

On the other hand, as described in Applicants' Background of the Invention, a conventional iterator function receives a set of arguments and returns a table to the SQL statement that invokes the function (e.g., Specification, page 1, paragraph 4). In particular, the iterator function creates a virtual table with a result set, and then the qualification (i.e., predicate) is applied to the virtual table to filter rows of data in the virtual table (e.g., Specification, page 3, paragraph 9). In many cases, only a small percentage of rows of data in the virtual table remain after the qualification is applied, but, unfortunately, because the qualification is applied after the rows of data are retrieved for the result set, many rows of data are unnecessarily retrieved for the

virtual table (e.g., Specification, page 3, paragraph 9). For example, an application program submitting SQL statement (2) is interested in articles with a score (assigned by the text\_search() iterator function) greater than or equal to 0.9, but, because the text\_search() iterator function is not aware of the score qualification, the text\_search() iterator function returns a virtual table with all articles that have "Bush" in the same sentence as "recession," without applying the predicate for the score (e.g., Specification, page 3, paragraphs 10-11).

To avoid this problem, under control of an iterator function processor, when an iterator function included in a statement is invoked, one or more predicates included in the statement are obtained and the one or more predicates are applied to a row of data before the row of data is returned.

The Examiner submits that the iterator function processor is anticipated by the sequence function compiler (Melton Abstract). Applicants respectfully traverse.

The Melton patent describes a SQL compiler that includes several features or sets of instructions that implement various aspects of the Melton invention (Col. 3, lines 65-67). The SQL compiler includes: an Offset function compiler 142, for compiling Offset sequence functions; a running and moving sequence function normalizer 144, which converts running and moving sequence functions into one or more Offset sequence functions that are used in conjunction with a set of auxiliary fields; and a Rows Since and THIS function normalizer 146, for converting a Rows Since sequence function into a normalized form that facilitates efficient computation of the Rows Since sequence function (Col. 4, lines 1-13).

Applicants respectfully submit that a SQL compiler does not anticipate the iterator function processor that executes as part of a data store engine and that, when an iterator function included in a statement is invoked, performs the claimed processing. For example, the compiler performs compilation tasks prior to execution of a function, whereas the iterator function processor processes an iterator function that is invoked.

The Examiner submits that applying the one or more predicates to a row of data is anticipated by "a comparison between the current row's values". Applicants respectfully traverse.

According to the Melton patent, sequence functions are functions that operate on ordered sets of rows and that require knowledge of or access to past values (Col. 1, lines 13-15). The purpose of the Rows Since function is to determine the distance, in rows, between the current

row and the most recent row in which a search condition was true (Col. 12, lines 37-40). The Rows Since sequence function (specifically when it contains a THIS sequence function) involves a comparison between the current row's values and selected values in each history row (Col. 13, lines 47-50). On the other hand, the claimed iterator function processor obtains one or more predicates included in the statement and applies the one or more predicates to *a row of data*. Such application to a single row of data is not anticipated by a Rows Since sequence function that compares values between rows.

Thus, claims 1, 14, and 27 are not anticipated by the Melton patent.

Dependent claims 2-5, 15-18, and 28-31 incorporate the language of independent claims 1, 14, and 27 and add additional novel elements. Therefore, dependent claims 2-5, 15-18, and 28-31 are not anticipated by the Melton patent for at least the same reasons as were discussed with respect to claims 1, 14, and 27.

Also, claim 2 describes that, under control of an iterator function processor, when an iterator function included in a statement is invoked (from independent claim 1), obtaining the one or more predicates comprises obtaining a qualification descriptor that describes the one or more predicates and one or more functions that are used to process the one or more predicates. The Melton patent at Col. 13, line 66 – Col. 14, line 25, is cited as anticipating this. Applicants respectfully traverse. The Melton patent describes parsing the Rows Since predicate to form a predicate tree (Col. 13, lines 65-67). Applicants respectfully submit that a predicate tree does not anticipate the claimed qualification descriptor that describes both predicates and functions that are used to process the predicates.

Claims 6, 19, and 32 describe under control of a data store engine, receiving a statement including an iterator function and one or more predicates, creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, and invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine (e.g., Specification, page 11, paragraph 34).

Applicants respectfully submit that the Melton patent does not anticipate an iterator function applying at least one predicate and the data store engine applying one or more of the predicates included in the statement that have not been applied by the iterator function. In addition, as discussed above, Applicants respectfully submit that the Melton patent does not anticipate the claimed qualification descriptor.

Thus, claims 6, 19, and 32 are not anticipated by the Melton patent.

Dependent claims 7-11, 20-24, 33-37 incorporate the language of independent claims 6, 19, and 32 and add additional novel elements. Therefore, dependent claims 7-11, 20-24, 33-37 are not anticipated by the Melton patent for at least the same reasons as were discussed with respect to claims 6, 19, and 32.

Also, claim 8 describes that, when the iterator function is invoked, receiving an indication from the iterator function indicating which of the one or more predicates were applied by the iterator function (e.g., Specification, page 13, paragraphs 38-42). The Examiner cites "testing predicate to be true" as teaching this. Applicants respectfully traverse. Receiving an indication from the iterator function indicating which of the one or more predicates were applied by the iterator function is not anticipated merely by testing a predicate to be true. For example, paragraph 39 of page 13 of the Specification describes that the iterator function processor applies some predicates and sends indicators to the data store engine to apply the remaining indicators, and paragraph 42, page 13 of the Specification describes that, by setting the one or more indicators, the iterator function processor is able to notify the data store engine which predicates have been applied so that the data store engine can save resources and avoid applying these.

Moreover, claims 10 and 11 describe applying one or more additional predicates to the received row of data. The Examiner cites Col. 14, lines 1-46 and lines 20-40, respectively, as anticipating claims 10 and 11. Applicants respectfully traverse. The cited portion of the Melton patent describes processing by the Rows Since sequence function and does not address the interaction of the claimed iterator function and the data store engine in applying predicates.

Claims 12, 25, and 38 describe the interaction between a data store engine and an iterator function processor in processing an iteration function. Under control of a data store engine, a statement including an iterator function and one or more predicates is received, a qualification descriptor is created that describes the one or more predicates and one or more functions that are

to be used to evaluate the one or more predicates, and the iterator function is invoked. Under control of an iterator function processor, the qualification descriptor is retrieved, a row of data that matches at least one of the one or more predicates in the qualification descriptor is obtained; indicators are set to indicate which of the one or more predicates have been applied by the iterator function processor (e.g., Specification, page 13, paragraph 39); and the row of data is returned to the data store engine. Under control of the data store engine, any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor are applied (e.g., Specification, page 11, paragraph 34).

Applicants respectfully submit that the Melton patent does not anticipate the subject matter of claims 12, 25, and 38.

Dependent claims 13, 26, and 39 incorporate the language of independent claims 12, 25, and 38 and add additional novel elements. Therefore, dependent claims 13, 26, and 39 are not anticipated by the Melton patent for at least the same reasons as were discussed with respect to claims 12, 25, and 38.

#### Conclusion

For all the above reasons, Applicants submit that the pending claims 1-39 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact her at (310) 553-7973 if the Examiner believes such contact would advance the prosecution of the case.

Dated: November 22, 2006

By:    /Janaki K. Davda                    

Janaki K. Davda  
Registration No. 40,684

Please direct all correspondences to:

David Victor  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: 310-553-7977  
Fax: 310-556-7984